$$
Here's the main idea: **R (and ggformula) can't directly facet on a purely numeric (continuous) variable**, so you first have to convert "Ring" into categories (bins). In other words, you "cut" the Ring variable into intervals. Then you can facet on those intervals.

Below is an example using **gf_histogram()** and **gf_facet_wrap()** with the `cut_number()` function, which divides "Ring" into a chosen number of equal-sized groups (by count, i.e., quantiles). You could also use `cut_width()` or `cut()` if you want to control the actual numeric breakpoints.

```
# Example: Facet histograms of Thumb by binned Ring length
gf_histogram(~ Thumb, data = Fingers, binwidth = 0.5) %>%
  gf_facet_wrap(~ cut_number(Ring, n = 4))
```

How it works

1. `gf_histogram(~ Thumb, data = Fingers, binwidth = 0.5)`
   Plots a histogram of the "Thumb" variable from the "Fingers" dataset with bin width 0.5.

2. `%>% gf_facet_wrap(~ cut_number(Ring, n = 4))`

   - `cut_number(Ring, n = 4)` splits the numeric "Ring" variable into 4 roughly equal-sized groups based on the number of observations.
   - `gf_facet_wrap()` then creates separate histogram panels (facets) for each of these groups.

If you prefer a different binning strategy, try:

- `cut_width(Ring, width = 0.5)` to split "Ring" into fixed-width intervals of 0.5 units each.
- `cut(Ring, breaks = c(5, 6, 7, 8, 9, 10))` (for example) to manually specify the breakpoints.

Either way, **the crucial point** is that you must turn "Ring" (continuous) into a categorical factor (via `cut*()` functions) before you can facet on it. Once you do that, you'll see how the distribution of "Thumb" changes across different ranges of "Ring."
$$